(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 June 2001 (28.06.2001)

PCT

(10) International Publication Number
WO 01/46800 A2

(51) International Patent Classification[7]: G06F 9/00

(21) International Application Number: PCT/US00/34458

(22) International Filing Date:
19 December 2000 (19.12.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/471,754 23 December 1999 (23.12.1999) US

(71) Applicant *(for all designated States except US)*: GENERAL INSTRUMENT CORPORATION [US/US]; 101 Tournament Drive, Horsham, PA 19044 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: CANDELORE, Brant [US/US]; 10124 Quail Glen Way, Escondido, CA 92029 (US). SPRUNK, Eric, J. [US/US]; 6421 Cayenne Lane, Carlsbad, CA 92009 (US).

(74) Agent: LIPSITZ, Barry, R.; Law Offices of Barry R. Lipsitz, Building No.8, 755 Main Street, Monroe, CT 06468 (US).

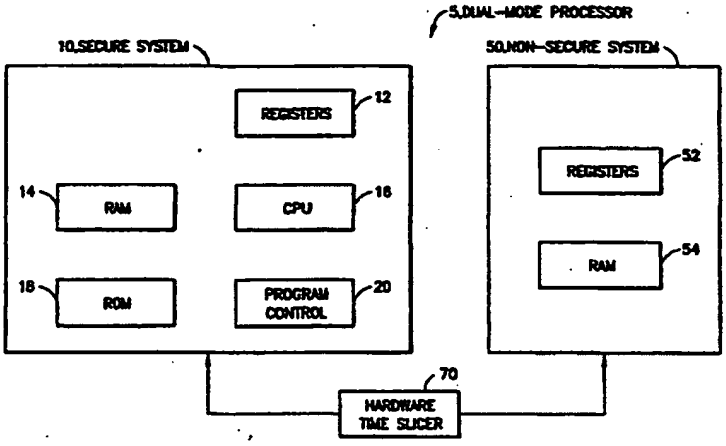
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: DUAL-MODE PROCESSOR



(57) Abstract: A multiple-mode processing circuit, such as a dual-mode processor (5), operates in at least first and second modes according to a switch (10). When a mode is active, data transfer between the processor and a respective memory occurs. Thus, instructions from the memory can be executed at the processor, and the results can be stored in the respective memory. For example, first and second memories (14, 54) may be provided for the first and second modes (10, 50), respectively. The memories are separate, and no data transfer can occur between the memories directly or via the processor. The first mode (10) may be a secure mode for secure processing operations, such as providing conditional access for television programming services at a set-top subscriber terminal. The second mode (50) may be a non-secure mode, such as for providing any other application at the terminal, e.g., program guide, shop at home service, etc. In one embodiment, a data bus is provided for time-multiplexed transfer of data between the processor and the respective memories. In another embodiment, switching of individual internal registers and external elements such as address and data latches, is provided.

WO 01/46800 A2

DUAL-MODE PROCESSOR

BACKGROUND OF THE INVENTION

The present invention relates to a circuit having a multi-mode processor, such as a dual-mode processor. The processor is particularly suitable for providing secure access control in a digital terminal for a subscriber television network.

In the present marketplace, economic pressures exist to add functionality and reduce the cost of consumer electronic products. This is particularly true for products such as the set-top terminal, also referred to as Integrated Receiver-Decoder (IRD) or subscriber terminal, which receive and decode television signals for presentation by a television. The signals can be delivered over a satellite, through a cable plant, or by means of terrestrial broadcast, for example.

One driver of the overall cost is the number of components that make up the product. One way to reduce the number of components is to combine functions that were normally performed by two or more Integrated Circuits (ICs) into one IC. This applies to dies which have embedded processors. For example, dies having both a memory and a Central Processing Unit (CPU) are now available.

Moreover, another way to increase functionality is through more efficient utilization of the circuit elements. The ever-increasing clock speeds of microprocessor implementations allowing more and more

processing to be achieved in the same amount of time by the same number of components. This allows products to have more features and be more responsive to the consumer. Increased clock speeds also allow a single processor to handle multiple applications.

By combining applications that run in multiple processors so that they may run in a single processor, the functionality of an individual circuit is increased, and component counts are lowered. However, combining applications is not a trivial task, especially with embedded processor applications, because the operating system, code structure, interrupt timing, and process interdependencies are often all changed when applications are merged. The embedded code with the combined functionality usually must be completely redesigned.

In personal computers, sophisticated operating systems like Windows(tm) can have multiple applications, usually non-interfering, that run simultaneously. But, the personal computer environment has a great deal of standardization both in terms of the hardware and software. Programmers can write applications expecting to use operating system services routines and common hardware. Embedded applications, however, typically do not benefit from either a standard operating system or for that matter a common hardware platform. But, even in the well-developed operating system environment of the personal computer, an application can fail, thus causing the entire system to stall or "hang", which necessitates the rebooting or resetting of the entire system. It is also possible

for a poorly written program or a malicious program like a "virus" to overwrite another application in memory. These virus programs can typically circumvent any memory management partitioning available to the operating system.

Operating systems have developed protection mechanisms to prevent programs which operate in a user application mode from overwriting privileged system information. These systems often use a logical circuit known as a Memory Management Unit (MMU) to prevent the host program executing in memory from being overwritten by a rogue application running in user mode. In such systems, there is no protection between applications executing simultaneously. Also, global space is shared by all tasks to avoid unnecessary replication of system service routines and to facilitate shared data and interrupt handling. Also, structures such as the stack may be accessible no matter what mode a program is running in. Application code might have access to some of the low-level general purpose and dedicated registers that the host program is using. And, such application programs may have access to stored system files.

Accordingly, it would be desirable to provide a way to combine disparate embedded applications with greater ease and without having to do a "re-design" of the code.

It would also be advantageous to provide a more secure scheme to allow independent sets of programs to execute on a microprocessor system without interference from each other.

The system should provide a dual-mode processor with secure and non-secure processing modes.

The system should be implementable in a subscriber terminal in a television network.

The present invention provides a system having the above and other advantages.

SUMMARY OF THE INVENTION

A multiple-mode processing circuit, such as a dual-mode processor, operates in at least first and second modes according to a switch. When a mode is active, data transfer between the processor and a respective memory occurs. Thus, instructions from the memory can be executed at the processor, and the results can be stored in the respective memory. For example, first and second memories may be provided for the first and second modes, respectively. The memories are separate, and no data transfer can occur between the memories directly or via the processor.

The first mode may be a secure mode for secure processing operations, such as providing conditional access for television programming services at a set-top subscriber terminal. The second mode may be a non-secure mode, such as for providing any other application at the terminal, e.g., program guide, shop at home service, etc.

In one embodiment, a data bus is provided for time-multiplexed transfer of data between the processor and the respective memories. In another embodiment, switching of individual internal registers and external elements such as address and data latches, is provided.

The processor may be switched between modes based on different schemes, including a fixed ratio of clock cycles, a fixed ratio of instructions executed at the processor, and respective priorities of the modes.

Moreover, the first and second modes may have different respective operating systems.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates an overview of a dual-mode processing circuit in accordance with the present invention.

FIG. 2 illustrates a dual-mode processor with a bus muxing circuit in accordance with the present invention.

FIG. 3 illustrates a dual-mode processor with switching for individually controlling external components in accordance with the present invention.

FIG. 4 illustrates a switch for selecting first and second modes of a dual-mode processor in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The present invention relates to a multiple-mode processor, such as a dual-mode processor.

FIG. 1 illustrates an overview of a dual-mode processor in accordance with the present invention.

In an example embodiment, the dual-mode processor 5 includes a secure system portion 10 and a non-secure system portion 50. The secure system 10 includes registers 12, a Random Access Memory (RAM) 14, a CPU 16, a Read-Only Memory (ROM) 18 and a program control function 20. The RAM 14 may store secure code and cryptographic keys, for example, for providing conditional access at a set-top terminal. The non-secure system 50 includes a register 52 and a RAM 54.

The processor 5 provides two functions (e.g., secure and non-secure functions) with absolute isolation between the functions. That is, each system 10, 50 has its own RAM 14, 54, so not a single bit of the secure system 10 is passed (e.g., leaks) to the RAM 54, and not a single bit of the non-secure system 50 is passed to the RAM 14. Thus, the data in the RAM 14 of the secure system 10 remains secure since data cannot be retrieved from it, or provided to it, by the non-secure system 50, either directly or via the CPU 16.

A hardware time slicer 70 is a switching means that can provide a 50% duty cycle by allocating half the time at the CPU 16 for processing data from the RAM 14 of the secure system 10, and the other half of the time for data from the RAM 54 of the non-secure system 50. With a 50% duty cycle, a 50 MHz processor would

look like two 25-MHz processors. Note that a commutator may alternatively be used.

Moreover, each system 10, 50 can have a different operating system.

FIG. 2 illustrates a dual-mode processor with a bus muxing circuit in accordance with the present invention.

Like-numbered elements correspond to one another in the figures.

The dual-mode processor 100 includes an internal data bus 105, an instruction queue 110, an instruction decoder and machine cycle encoder 115, and a timing and logic control function 120, which includes a program mode data bus, hold, interrupt, wait, write sync, control clocks and other control functions.

Also provided are a data buffer 125, a program model selector 130, and interrupt control 135, and a mode timing function 140. As shown, the invention duplicates certain context registers so that the reset of the processor may be used in two independent modes of operation. The program mode selector 130 allows switching between the modes.

A register bank multiplexer 150 includes two banks of registers. A register bank for a first mode (mode "A") of the dual-mode processor 100 includes general purpose registers 160, an index register 162, a stack pointer/return address register 164, a program counter 166, a memory management register 168, a cache controller register 170, and an interrupt control register 172.

Similarly, a register bank for a second mode (mode "B") of the dual-mode processor 100 includes general purpose registers 180, an index register 182, a stack pointer/return address register 184, a program counter 186, a memory management register 188, a cache controller register 190, and an interrupt control register 192.

Also provided are an address buffer 194 and a memory bank multiplexer 195, with a memory bank A 197 for use in the first mode, and a memory bank B 198 for use in the second mode.

In this bus muxing embodiment, data is time multiplexed for transport on the bus 105 during the respective modes.

FIG. 3 illustrates a dual-mode processor controlling external components in accordance with the present invention. The figure shows an example of using mode A/B switching internally and externally to an IC, such as an Application-Specific Integrated Circuit (ASIC) on which the dual-mode processor is provided. The address space can be split using modes A/B. Here, each individual internal register is switched, whereas in the embodiment of FIG 2, a bus muxing circuit is used.

A circuit, shown generally at 300, includes the dual-mode processor 305 and a number of external components, including address latches A (350) and B (352), and data latches A (354) and B (356). Processor mode A has example Port 1 (358), . . . , Port N (362), and processor mode B has example Port 1 (360), . . . ,

Port N (364). Memory A (197) and memory B (198) are also provided.

The dual-mode processor 305 includes the instruction decode and machine cycle encoder 115, a mode A/B time switcher 310 (see detail of switch in FIG. 4), address generators 315, 317, instruction pipelines 320, 322, data buffers 325, 327, registers 160, 180, caches 170, 190, and Memory Management Units (MMUs) 168, 188. The MMUs 168, 188 provide virtual-to-physical address translation.

Data and Instruction sections of memory may be defined. In addition, there may be control registers which allow the privileged user to selectively grant access to various memory blocks to an application running in User Mode.

A path 380 carries a Mode A/B selection signal that is provided by the mode A/B timer switcher 310 to indicate which mode is currently running.

The address generators 315, 317 provide memory addresses to the respective address latches 350, 352, which latch addresses for reading/writing operations at the memories 197, 198, respectively.

The data buffers 325, 327 send and receive data to and from the data latches 354, 365 and the ports 358, 360, . . . , 362, 364.

The present invention allows two or more disparate sets of programs to be run by a single microprocessor with complete independence and 100% isolation between them. For this discussion, we define independent sets of programs "A" and "B", respectively, but the concept can easily be extended to additional independent sets

of programs. Programs from one set cannot access programs from the other set nor influence the execution of programs from the other set. Each set of programs can be considered to have its own operating system with various application programs running concurrently. The operating system and application programs from one set cannot interfere with the other set.

Moreover, it is not possible for either the operating system program or application program in one set, even if written with malicious intent, to read out the other programs or to learn any details of the other programs running in the other set.

By having sets of programs A and B share functional hardware such as the Instruction Decoder and Machine Cycle Encoder circuitry 115, along with the Timing and Control Logic circuitry 120, more efficient use is made of the area of an IC chip, such as a Very Large Scale Integrated Circuit (VLSI) chip. In particular, by eliminating an entire microprocessor from a system by combining operations of set A programs working with set B programs, significant cost savings can be realized by reducing the overall part count.

The invention allows two (or more) sets of programs that previously may have ran on separate processors to work together on a single processor with little or no changes to either of the original sets of programs. This results in a significant savings in code development time and expense. Moreover, even if both programs were combined to form one larger program with the combined functionality, and this program was executed out of the same type of processor, the new

larger program would still need to be completely checked out and debugged. The present invention avoids this problem by allowing the two smaller program to exist as before.

In one possible embodiment, one set of programs can be considered "secure code" executing cryptographic routines that perform access control functions in a set-top box. This set of programs is not interfered with by programs executing in the other "non-secure" set of programs. In a set-top box, this allows the main microprocessors to be combined into one device.

"Access control" refers to algorithms running in a "secure" processor used to determine whether or not a decoder is authorized to view a particular program. The program may be given away, needing a subscription, or needing the user to make a purchase.

FIG. 4 illustrates a switch for selecting Mode A or B of the dual-mode processor in accordance with the present invention.

The switch 400 is a negative edge-triggered D-type flip-flop. The flip-flop 405 receives a master clock signal via a line 410, and outputs either a Mode A select signal on a line 420, or a Mode B select signal on a line 430. The master clock signal may be used for other purposes via line 440.

The processor requires a switch means for switching between the running of one set of programs to running the other set of programs. Time can be a means for switching between sets of applications. The following are some options:

1. A fixed ratio of clock cycles. If set to 50 - 50 %, then every other clock cycle would be used for Set A (Mode A), and every other cycle for Set B (Mode B) programs. If set to 80 - 20 %, then Set A would execute for four clock cycles, and then Set B for one cycle.

2. A Fixed number of clock cycles in a row. If set to 50 - 50%, then Set A would get ten clock cycles in a row, and Set B would get ten clock cycles in a row. If set to 80 - 20%, then Set A would get sixteen clock cycles in a row, and Set B would get four clock cycles in a row.

3. A fixed ratio of instructions executed. If set to 50 - 50%, then Set A and Set B get every other instruction executed. If set to 80 - 20%, then Set A would get four instructions, and then Set B would get one instruction.

4. A fixed number of instructions in a row. If set to 50 - 50%, then Set A would execute ten instructions in a row, and Set B would execute ten instructions in a row. If set to 80 - 20%, Set A would execute sixteen instructions in a row, and Set B would execute four instructions in a row.

5. Dynamic clock or instruction allocation is possible while guaranteeing minimum clock or instruction execution. It is possible for a process that is executing a low priority routine to give up a certain number of clock or instruction cycles to the other process. When a high-priority process is invoked, e.g., an interrupt, then the pre-empted

process can get back the clock or instruction cycles it gave up.

The following optional implementations are also envisioned:

1. Both Set A and Set B circuits can be clocked at the same time. The clock and instruction cycle guarantees are for gaining access to any shared resource. In the scenario described above, the shared resources are: Instruction Decoder and Machine Cycle Encoder 115, and the Timing and Control Circuitry 120. They would follow a strict clock and instruction allocation scheme.

2. Memory Space can be completely separate between Set A and Set B programs.

3. A requirement can be placed on the CPU that the minimum timing can never be violated. This may necessitate additional duplication of hardware. Or, it may simply allow completion of a lengthy instruction, giving priority of the next instruction to the other set of programs.

A consequence of the invention is that any data dealing with the context or state of what and where the microprocessor system was executing in its memory must be duplicated for each independent set of programs executing in the microprocessor system.

The processor of the present invention has the following structures, which provide memory and, therefore, context to a system: a stack and a stack pointer 164, 184 or at least a register containing a return address, general purpose registers 160, 180, program counter 166, 186, CPU status register, MMU

control registers 168, 188, Cache control registers 170, 190, various I/O registers, and interrupt controller. In addition, both the internal and external RAM and ROM of the system may be switched. This is conceptually similar to switching banks of memory. If the most significant bit of the address is a function of the mode of which set of programs was executing (A or B), then, when high, the upper memory bank 197 can be used for set of program A, and when low, the lower memory bank 198 can be used for set of programs B.

If the microprocessor system has internal cache, then it may or may not be required to duplicate this structure depending on whether or not this memory can be read out discretely by the CPU. Any cache control registers would need to be duplicated. More efficient use of the cache structure can be achieved by duplicating it because each set of programs will be operating in different memory spaces. A higher cache hit rate will be achieved with a duplicate cache since two sets of independent programs will be executing.

Each program set has its own boot program. The boot program can set the parameters which, for example, identify which blocks of code are low priority and which ones are high priority and might require the full allocated number of clock cycles.

The invention is useful in television set-top terminals (e.g., decoders) in eliminating the need for a separate processor to handle Access Control and Cryptography. The processor that is used to run the

decoders can run this function as well using the dual mode feature.

In optional uses of the invention, it is possible to try out some unknown code. For example, the Internet allows many small Java application routines or "applets" of unknown quality and origin to be downloaded from various Web sites. The dual-processor system of the present invention (e.g., as implemented in a Web-compatible set-top box, can provide protection while running this code. This may be a way to keep applications from interfering with each other.

In a further option, the invention can be used to implement a type of firewall, e.g., to isolate data processing spaces.

In a further option, the invention can be used to implement a fault tolerant computer, e.g., a personal computer system that does not fail (crash) when the primary processor crashes.

Accordingly, it can be seen that the present invention provides a dual or other multiple-mode processor. The processor time-shares among different processes by allocating its time for executing instructions. In a particular embodiment, the processor includes secure and non-secure systems that store data, and retrieve data from, separate memories. The processor may be used, e.g., to provide conditional access to television programming services.

Although the invention has been described in connection with various specific embodiments, those skilled in the art will appreciate that numerous adaptations and modifications may be made thereto

without departing from the spirit and scope of the invention as set forth in the claims.

For example, the invention is suitable for use with virtually any type of network, including cable or satellite television broadband communication networks, local area networks (LANs), metropolitan area networks (MANs), wide area networks (WANs), internets, intranets, and the Internet, or combinations thereof.

Additionally, known computer hardware, firmware and/or software techniques may be used to implement the invention.

What is claimed is:

1. A multiple-mode processing circuit, comprising:

a processor operating in at least first and second modes;

timing means for switching the processor between the at least first and second modes;

a first memory for providing data to, and receiving data from, the processor while in the first mode;

a second memory separate from said first memory for providing data to, and receiving data from, the processor while in the second mode; and

means responsive to said timing means for managing a transfer of data between the processor and the first memory, and between the processor and the second memory.

2. The circuit of claim 1, wherein:

the first mode is a secure mode for secure processing operations, and the second mode is a non-secure mode for non-secure processing operations; and

said managing means prevents the transfer of data from the second memory to the first memory, or from the first memory to the second memory.

3. The circuit of claim 2, wherein:

the secure processing operations comprise providing conditional access for television programming services.

4. A television set-top terminal that comprises the circuit of claim 3.

5. The circuit of claim 1, wherein:

said managing means comprise a data bus for time-multiplexed transfer of data between the processor and the first memory, and between the processor and the second memory.

6. The circuit of claim 5, wherein:

the processor and data bus are provided in an Integrated Circuit (IC).

7. The circuit of claim 1, wherein:

said managing means comprise first and second registers that are activated in response to said switching to transfer data between the processor and

the first memory, and between the processor and the second memory.

8. The circuit if claim 7, wherein:

said processor and first and second registers are provided in an Integrated Circuit (IC).

9. The circuit of claim 8, wherein:

said first and second memories are external to said IC; and

said managing means comprises first and second latches that are external to said IC, and that are activated in response to said switching to transfer data between the processor and the first memory, and between the processor and the second memory.

10. The circuit of claim 1, wherein:

said timing means switches the processor between the at least first and second modes according to a fixed ratio of clock cycles.

11. The circuit of claim 1, wherein:

said timing means switches the processor between the at least first and second modes according to a fixed ratio of instructions executed at the processor.

12. The circuit of claim 1, wherein:

said timing means switches the processor between the at least first and second modes according to respective priorities of the at least first and second modes.

13. The circuit of claim 1, wherein:

the first and second modes have different respective operating systems.

14. The circuit of claim 1, wherein:

first and second applications are responsive to the first and second memories, respectively, for executing in said first and second modes, respectively.

15. The circuit of claim 14, wherein:

said first and second applications are disparate.

16. The circuit of claim 1, wherein:

first and second sets of programs that are independent from one another are responsive to the first and second memories, respectively, for

executing in said first and second modes, respectively.

17. The circuit of claim 1, wherein:

said managing means prevent any data from being transferred between said first and second memories.

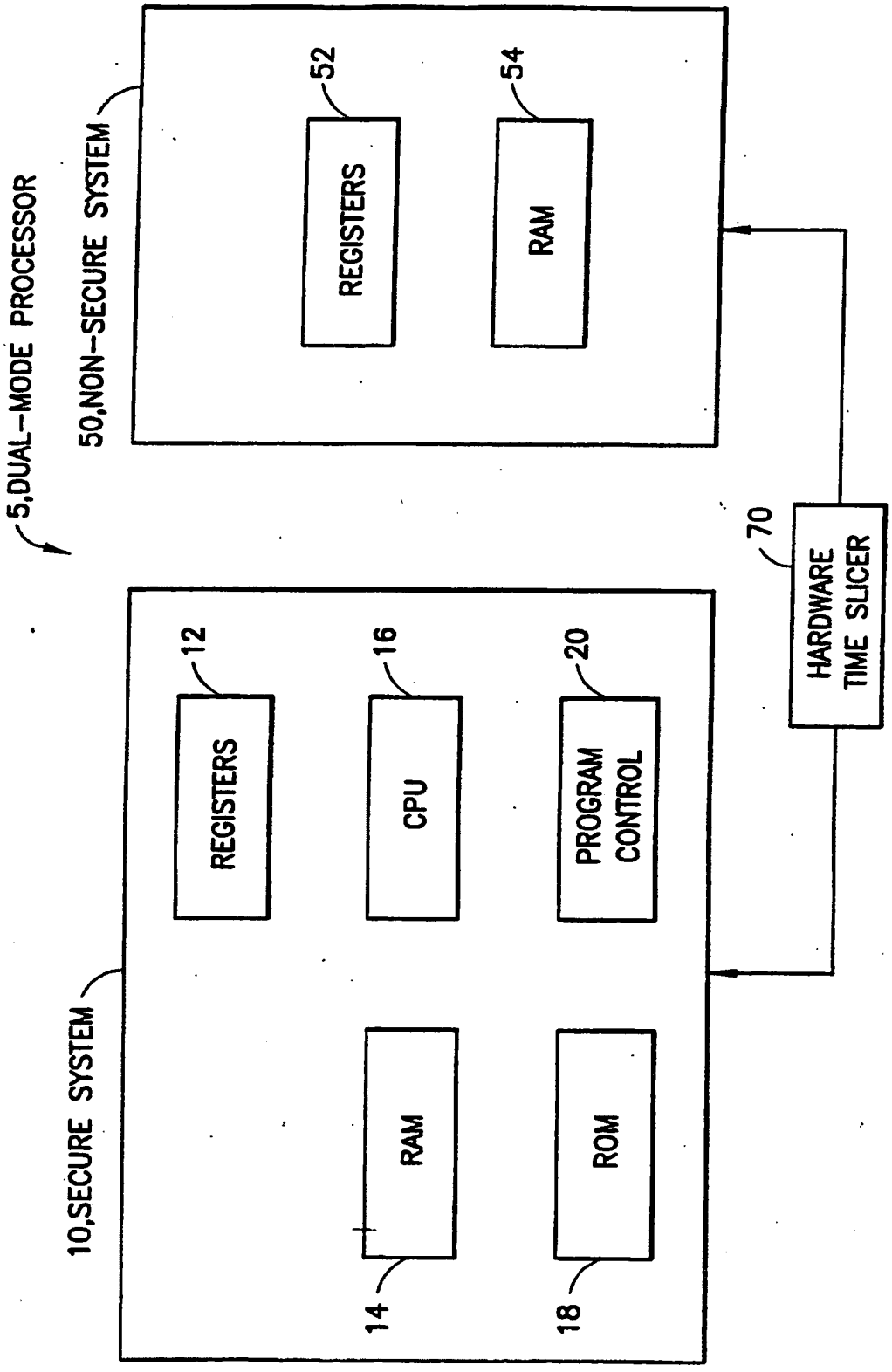

5,DUAL-MODE PROCESSOR
50,NON-SECURE SYSTEM
52
REGISTERS
54
RAM
70
HARDWARE TIME SLICER
10,SECURE SYSTEM
12
REGISTERS
16
CPU
20
PROGRAM CONTROL
14
RAM
18
ROM

FIG.1

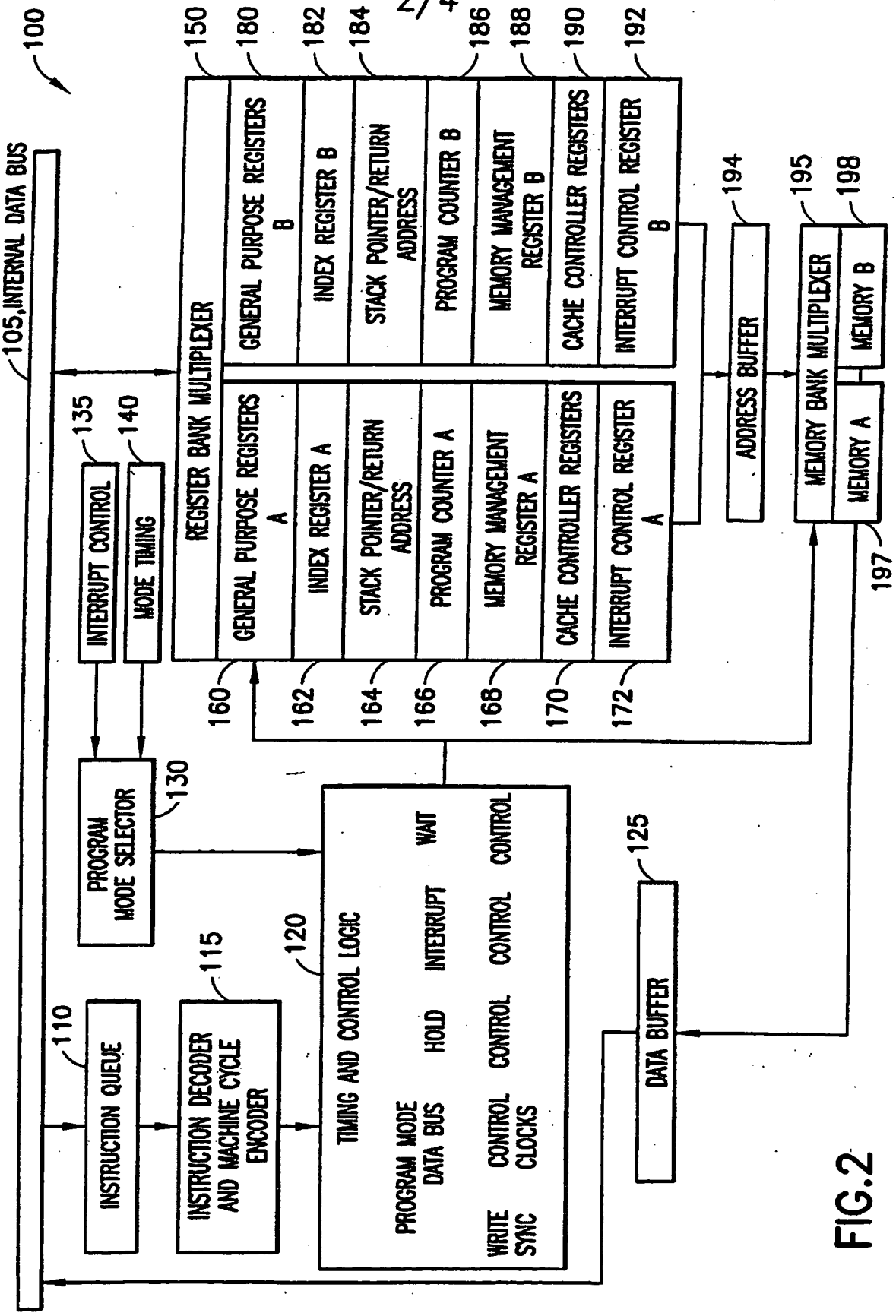

100
105, INTERNAL DATA BUS
150
180
182
184
186
188
190
192
194
195
198
197
REGISTER BANK MULTIPLEXER
GENERAL PURPOSE REGISTERS B
INDEX REGISTER B
STACK POINTER/RETURN ADDRESS
PROGRAM COUNTER B
MEMORY MANAGEMENT REGISTER B
CACHE CONTROLLER REGISTERS
INTERRUPT CONTROL REGISTER B
ADDRESS BUFFER
MEMORY BANK MULTIPLEXER
MEMORY B
MEMORY A
GENERAL PURPOSE REGISTERS A
INDEX REGISTER A
STACK POINTER/RETURN ADDRESS
PROGRAM COUNTER A
MEMORY MANAGEMENT REGISTER A
CACHE CONTROLLER REGISTERS
INTERRUPT CONTROL REGISTER A
160
162
164
166
168
170
172
135
INTERRUPT CONTROL
140
MODE TIMING
130
PROGRAM MODE SELECTOR
110
INSTRUCTION QUEUE
115
INSTRUCTION DECODER AND MACHINE CYCLE ENCODER
120
TIMING AND CONTROL LOGIC
PROGRAM MODE
DATA BUS
HOLD
INTERRUPT
WAIT
WRITE
SYNC
CONTROL
CLOCKS
CONTROL
CONTROL
CONTROL
125
DATA BUFFER

FIG.2

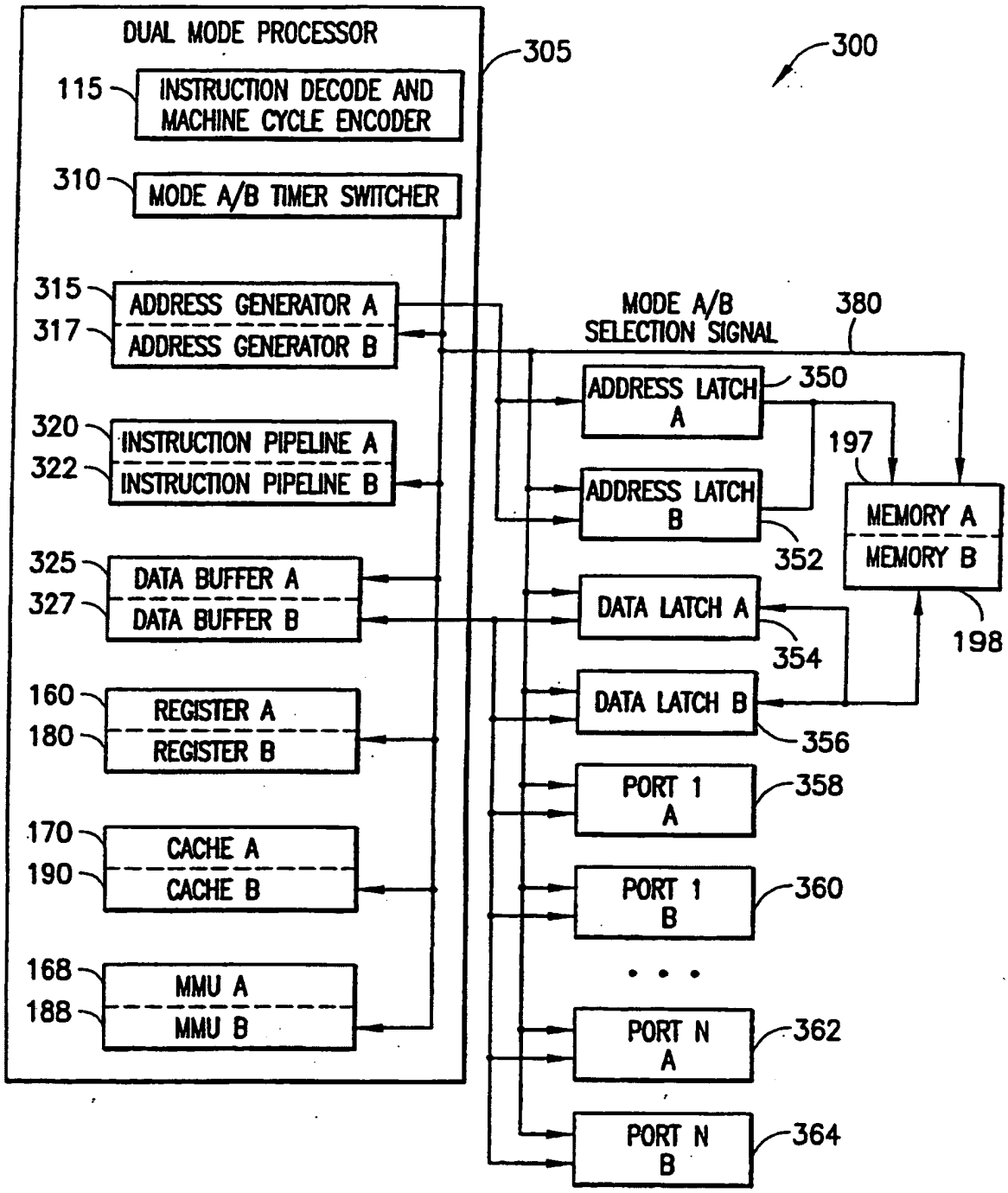

DUAL MODE PROCESSOR
305
300
115
INSTRUCTION DECODE AND MACHINE CYCLE ENCODER
310
MODE A/B TIMER SWITCHER
315
317
ADDRESS GENERATOR A
ADDRESS GENERATOR B
320
322
INSTRUCTION PIPELINE A
INSTRUCTION PIPELINE B
325
327
DATA BUFFER A
DATA BUFFER B
160
180
REGISTER A
REGISTER B
170
190
CACHE A
CACHE B
168
188
MMU A
MMU B
MODE A/B SELECTION SIGNAL
380
ADDRESS LATCH A
350
ADDRESS LATCH B
352
197
MEMORY A
MEMORY B
198
DATA LATCH A
354
DATA LATCH B
356
PORT 1 A
358
PORT 1 B
360
PORT N A
362
PORT N B
364

FIG.3

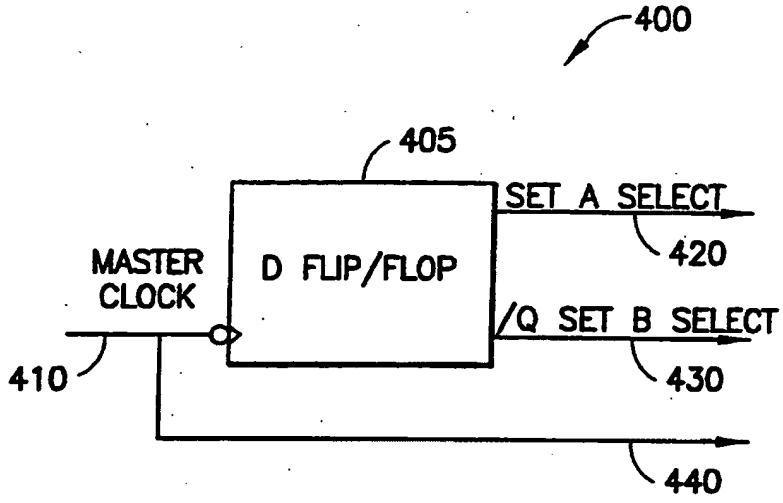

400
405
SET A SELECT
420
MASTER CLOCK
D FLIP/FLOP
/Q SET B SELECT
410
430
440

FIG.4